

Nokia Research Center / Helsinki  
Dirk Trossen, Dana Pavel

FINAL  
24.10.06

## **N-RSA HIGH-LEVEL SYSTEM ARCHITECTURE**

Authors: Dirk Trossen, Dana Pavel  
Nokia Research Center / Helsinki

Editor: Elena Balandina  
Nokia Research Center / Helsinki

### **ABSTRACT**

This document outlines the high level system architecture for the Nokia Remote Sensing Architecture (N-RSA). For that, overall architecture requirements are defined from which a high level component architecture is derived.

Version:	1.0
Authors:	Dirk Trossen, Dana Pavel

Nokia Research Center / Helsinki  
Dirk Trossen, Dana Pavel

FINAL  
24.10.06

**GLOSSARY**

SIP	Session Initiation Protocol
N-RSA	Nokia Remote Sensing Architecture
XML	Extensible Markup Language

Nokia Research Center / Helsinki  
Dirk Trossen, Dana Pavel

FINAL  
24.10.06

**TABLE OF CONTENTS**

**1. INTRODUCTION..... 4**

**2. OVERALL ARCHITECTURAL REQUIREMENTS ..... 4**

**3. ARCHITECTURE COMPONENTS ..... 4**

    3.1 APPLICATIONS ..... 5

    3.2 MIDDLEWARE ..... 5

    3.3 SENSOR API..... 8

    3.4 SENSOR PROTOCOLS ..... 8

    3.5 MULTISENSOR MODULE..... 8

    3.6 REPOSITORY..... 9

**4. REFERENCES..... 9**

Nokia Research Center / Helsinki  
Dirk Trossen, Dana Pavel

FINAL  
24.10.06

## 1. INTRODUCTION

This report describes the Remote Sensing Architecture (N-RSA) as a high-level component architecture. For each of the components in this architecture, we will define component-specific requirements.

Note that throughout the document, we will use the keywords MUST/SHOULD/MAY for the requirement definitions, as introduced in [1].

## 2. OVERALL ARCHITECTURAL REQUIREMENTS

The following section provides the main overall technical requirements of the N-RSA:

- Req. 1** The N-RSA MUST provide a repository for the entire data source assets.
- Req. 2** The N-RSA MUST provide collection of sensor data locally and remotely
  - Req. 2.a** It MUST be possible to obtain plain sensor data
  - Req. 2.b** It MUST be possible to obtain higher level semantic data
- Req. 3** The N-RSA SHOULD support
  - Req. 3.a** the definition of access rights for particular data, depending on the accessing entity
  - Req. 3.b** the (temporary) storage of data for later retrieval
- Req. 4** The N-RSA MUST support different communication media within the distributed system
- Req. 5** The architecture MUST allow for extensions
  - Req. 5.a** It MUST support the integration of new sensors
  - Req. 5.b** It SHOULD support the integration of new aggregation functionality, therefore enriching the availability of higher-level, derived data.

In the following, we will introduce a component architecture that implements the abovementioned high-level requirements. For each of the components, we will define component-specific requirements that will detail the above presented list.

## 3. ARCHITECTURE COMPONENTS

Figure 1 shows the N-RSA component-level system architecture that will be the basis of development throughout N-RSA.

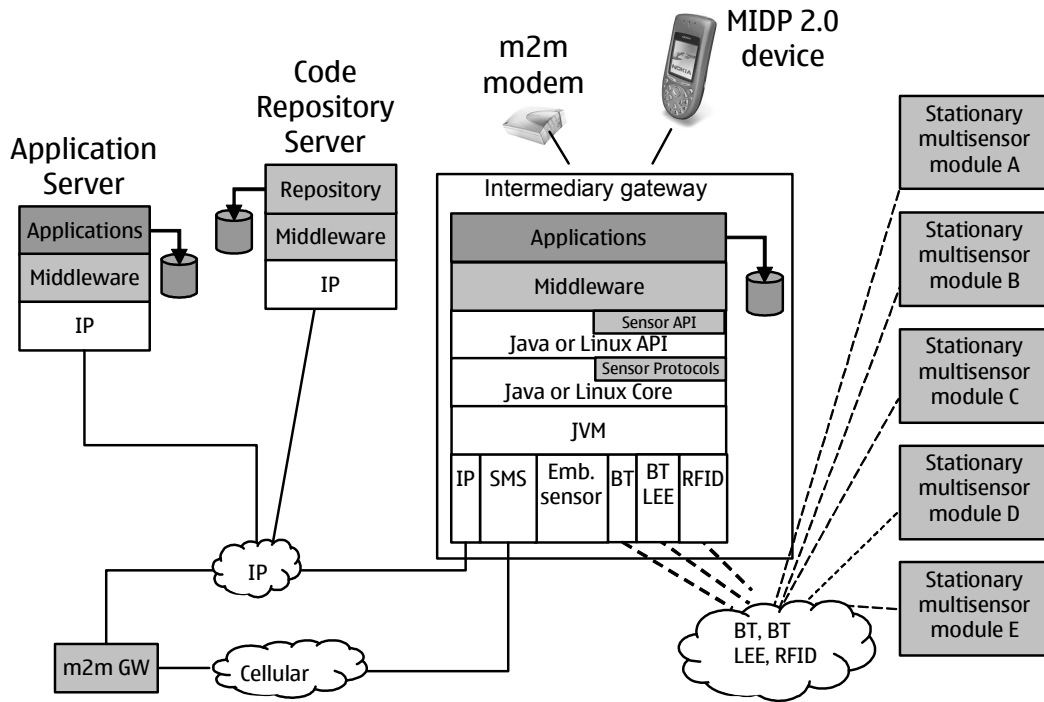


Figure 1 Component Architecture

In the following subsections, we will describe more detailed requirements for each of the components.

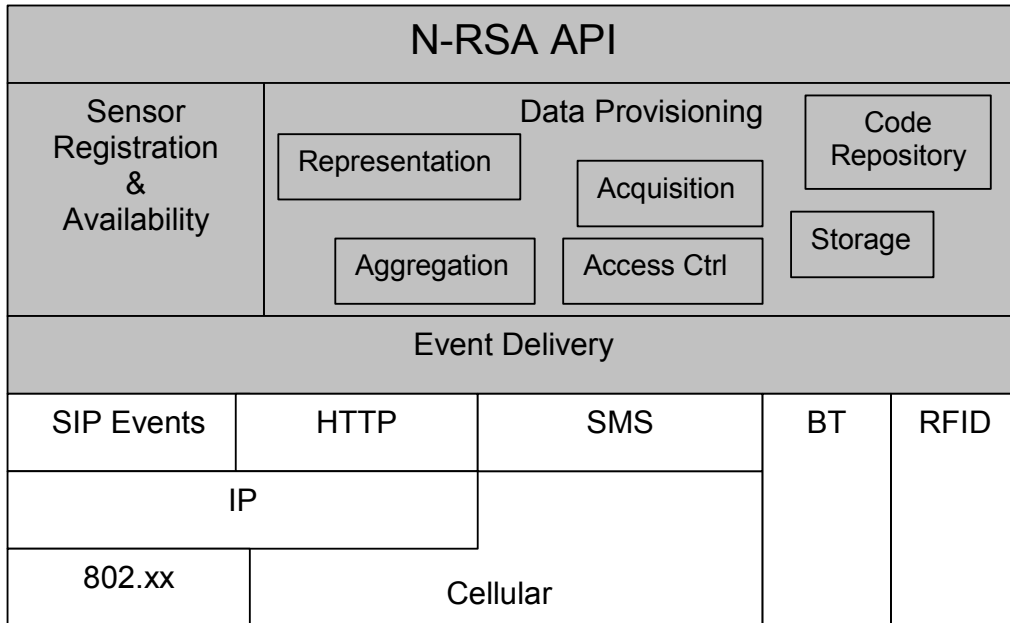
### 3.1 Applications

The **application** component implements the desired functionality for the *vertical application*. For that, it accesses the middleware component in order to implement the required remote data gathering functionality within the vertical application. Usually, the application will run on dedicated application servers and/or on the intermediary gateway (m2m modem or MIDP 2.0 device).

However, since the actual application functionality is not within the scope of our considerations, no requirements with respect to the N-RSA are to be defined.

### 3.2 Middleware

The **middleware** component mainly implements the overall system architecture requirements 1 to Req. 5 on software level. This is reflected by the sub-components as shown in Figure 2.



**Figure 2** Middleware Components

- Req. 6** The middleware **MUST** provide a sensor registration & availability component.
  - Req. 6.a** It **MUST** allow for registering data sources (sensors) with the middleware, together with its capabilities
  - Req. 6.b** It **MUST** allow for querying (or poll) for availability of registered data sources, constrained by given capabilities
  - Req. 6.c** It **MUST** allow for subscribing to the availability of current or future registered data sources, constrained by given capabilities, generating notifications to the subscriber with the matching data sources upon availability of the data sources
- Req. 7** The middleware **MUST** provide a data collection and provisioning component.
  - Req. 7.a** It **MUST** support local and remote sensors
  - Req. 7.b** It **MUST** support abstraction of data in a defined abstraction model.
    - Req 7.b.I** The abstraction model **MUST** support simple as well as aggregated sensor information
    - Req 7.b.II** The abstraction model **MUST** provide a metadata description (XML/RDF) of sensors and aggregated data

Nokia Research Center / Helsinki  
Dirk Trossen, Dana Pavel

FINAL  
24.10.06

- Req 7.b.III** Said abstraction model **MUST** allow for inclusion of new sensors and sensor types
- Req. 7.c** It **MUST** support acquisition of data.
  - Req 7.c.I** The acquisition **MUST** allow for query of updated values
  - Req 7.c.II** The acquisition **MUST** allow for subscription to updated values, generating notifications to the subscriber with said updated values
  - Req 7.c.III** The acquisition **MUST** provide a query language to formulate queries/subscriptions appropriately, based on the abstraction model
- Req. 7.d** It **MUST** support hierarchical provisioning of data, based on the abstraction model, in order to support aggregation of data
- Req. 7.e** It **SHOULD** allow for definition of access rights for particular data, depending on the accessing entity
- Req. 7.f** It **SHOULD** allow for (temporary) storage of data for later retrieval
- Req. 8** The middleware **MUST** provide an event delivery component
  - Req. 8.a** It **MUST** support arbitrary events plus event-specific payload
  - Req. 8.b** It **MUST** be bearer agnostic, i.e., it **MUST** support any bearer of information
    - Req 8.b.I** It **MUST** support at least SMS and Bluetooth
    - Req 8.b.II** It **SHOULD** also support IP-based bearers.
      - It **MUST** support HTTP
      - It **SHOULD** support SIP events [2]
    - Req 8.b.III** It **SHOULD** support RFID bearers
  - Req. 8.c** It **MUST** support bearer-specific addressing of remote components
    - Req 8.c.I** It **MUST** support phone numbers for SMS
    - Req 8.c.II** It **MUST** support IP addresses for HTTP
    - Req 8.c.III** It **MAY** support URI-like addressing for HTTP
    - Req 8.c.IV** It **MUST** support URI addressing for SIP events
    - Req 8.c.V** It **MUST** support the extension of new addressing
      - It **SHOULD** support new addressing schemes
      - It **MAY** support user defined addressing
- Req. 9** The middleware **MUST** allow for extensions

Nokia Research Center / Helsinki  
Dirk Trossen, Dana Pavel

FINAL  
24.10.06

- Req. 9.a** It MUST support the integration of new sensor types
- Req. 9.b** It SHOULD support a *code repository component*, allowing for the integration of new aggregation functionality, therefore enriching the availability of higher-level, derived data.

### 3.3 Sensor API

The **sensor API** component enables the access to the sensor protocol component. For that,

- Req. 10** It MUST implement appropriate functions to properly enable said access to the sensor protocol component by the upper layer middleware component.

### 3.4 Sensor Protocols

The **sensor protocols** component implements the communication with local and/or proximity sensors. In this, the sensor protocols component implements the lower layer functionality required to realize Req. 1 to Req. 5.

More specifically for this component, the following requirements apply:

- Req. 11** It MUST support local as well as proximity sensors
  - Req. 11.a** It MUST support Bluetooth proximity sensors
  - Req. 11.b** It SHOULD support RFID proximity sensors
- Req. 12** It MUST support acquisition of sensor data
  - Req. 12.a** It MUST support poll-like acquisition
  - Req. 12.b** It SHOULD support subscribe-like acquisition
- Req. 13** It MUST support discovery of sensors
  - Req. 13.a** It MUST support poll-like discovery
  - Req. 13.b** It SHOULD support subscribe-like discovery
- Req. 14** It MUST support adding new sensor types to the environments
- Req. 15** It SHOULD support setting and obtaining configuration data of sensor modules

### 3.5 Multisensor Module

The **multisensor module** implements the local sensing part of the overall system architecture requirements. It is not within the scope of this document to define more specific requirements for this component. However, general means for sensor discovery and sensor acquisition must be supported.

Nokia Research Center / Helsinki  
Dirk Trossen, Dana Pavel

FINAL  
24.10.06

### 3.6 Repository

The **repository** component implements the (optional) extendibility of the N-RSA with respect to Req. 5.b. The component applies the functionality of the middleware with respect to the integration of aggregation code during runtime, i.e., it uses the code repository sub-component of the N-RSA middleware (see Figure 2). The repository is likely to reside on a separate server (the *code repository server*, see Figure 1).

More specifically for this component, the following requirements apply:

- Req. 16** It MUST allow for uploading new code modules to the repository (enriching the repository)
  - Req. 16.a** It MUST provide permanent storage facilities for the uploaded code modules
- Req. 17** It MUST allow for uploading new code modules to the application server or to the intermediary gateway (usage of code)

## 4. REFERENCES

- [1] S. Bradner, "Key words for use in RFCs to indicate requirement levels," RFC 2119, Internet Engineering Task Force, March 1997
- [2] A. Roach, "SIP-Specific Event Notification", RFC 3265, July 2002